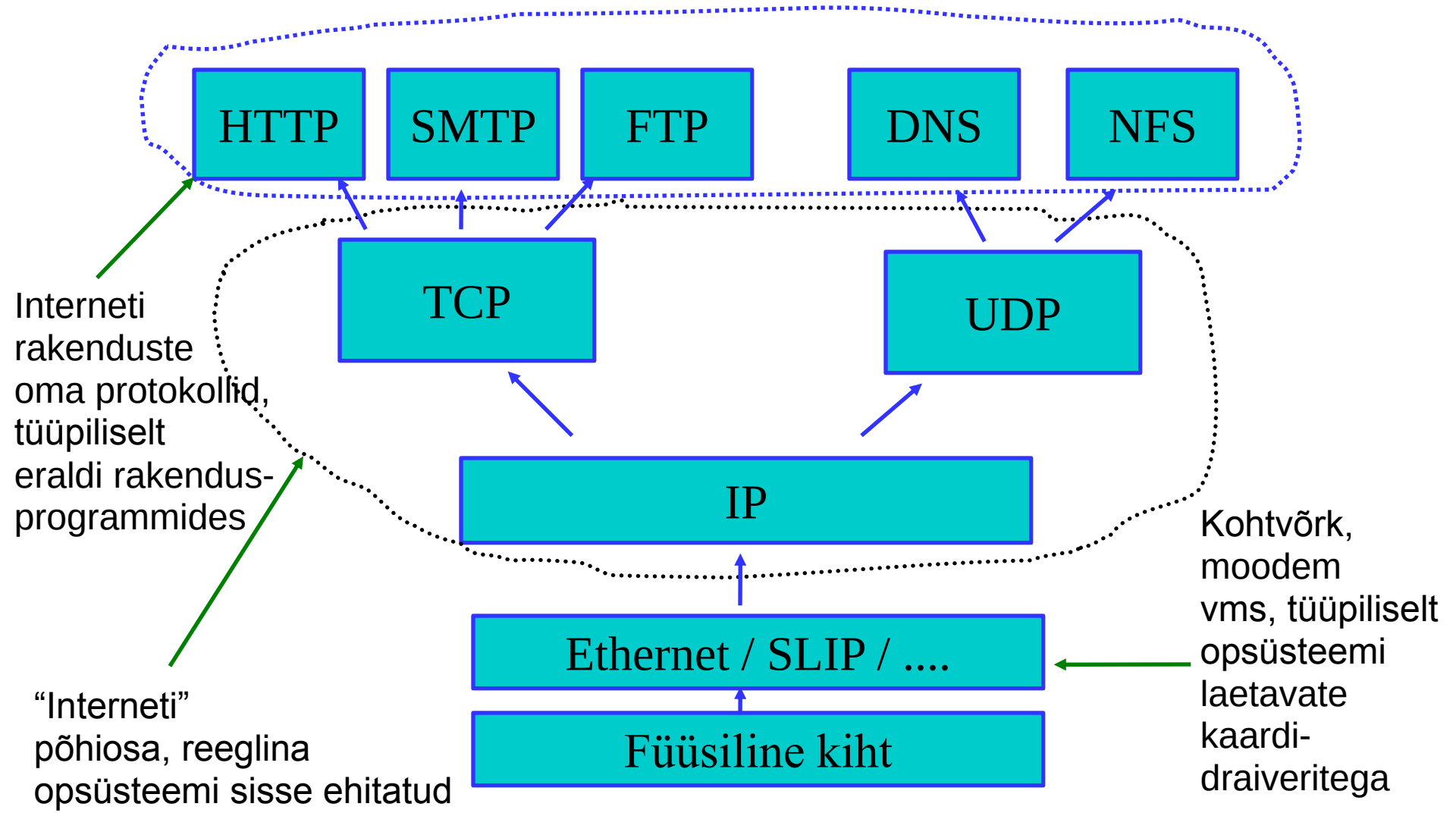
Sissejuhatus infotehnoloogiasse

interneti rakenduste tehnoloogia

Loengu ülevaade. Interneti rakendused.

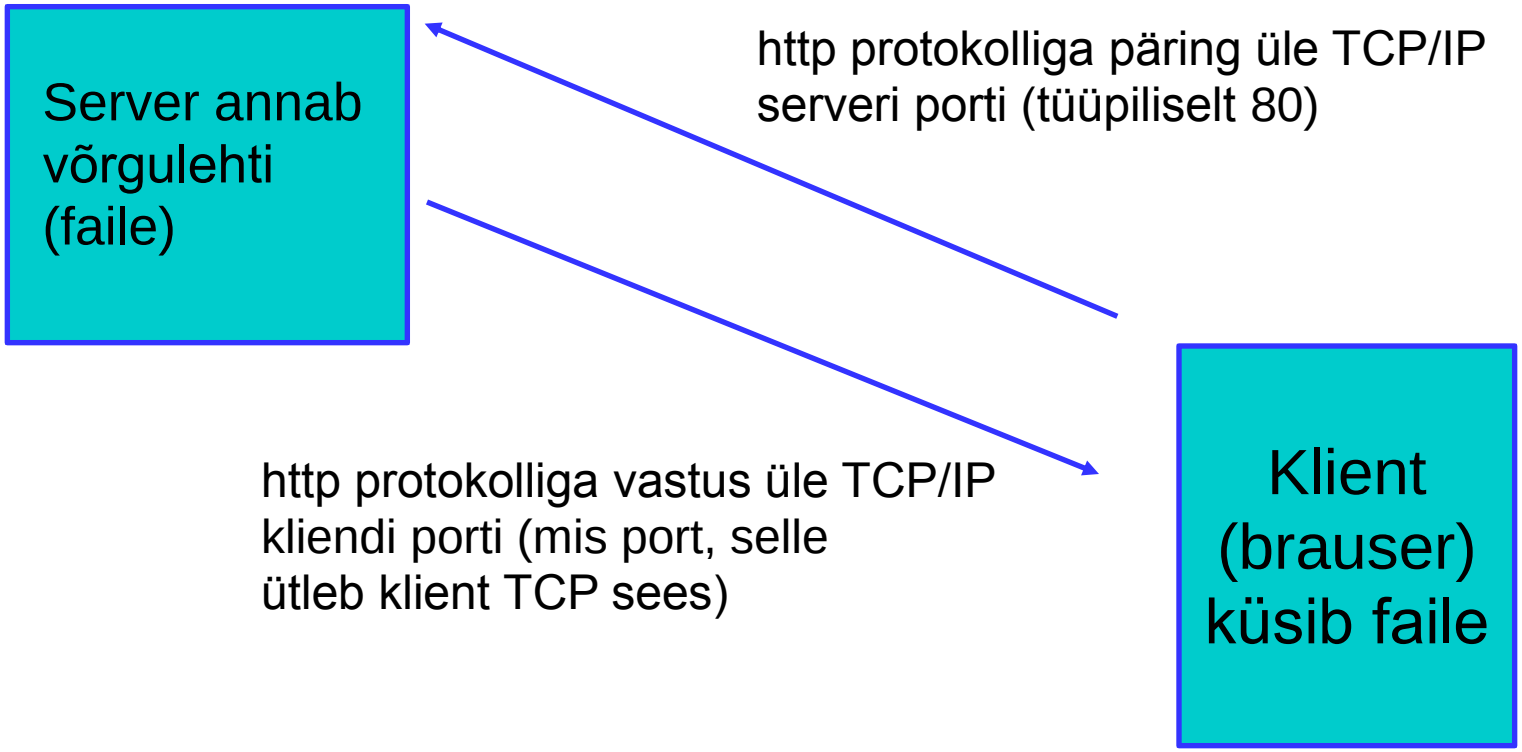
- Meeldetuletus: baasprotokollid, HTTP protokoll
- Failide tõmbamine ja serveerimine:
  - Kuidas http abil faili tõmmata
  - Kuidas http abil faili serveerida
  - Info saatmine serverile: cgi protokoll
- Mis brauseril sees on:
  - Html, css, javascript: baastehnoloogiad
  - Xml, xslt ja veel hulk eritehnoloogiaid
- XML ja JSON
- Semantika, andmete esitus, rdf ja ontoloogiad

Kokkuvõte protokollindusest internetis



HTTP ühendused: failide küsimine ja nende andmine

- **HTTP on omaette protokoll**, mida kasutatakse veebilehtede, piltide, tekstifailide, zip failide jne jne saatmiseks veebiserveri ja brauseri vahel.



Kuidas html-faili tõmmata?

- Vaatame lihtsat java-koodi näidet:

http://java.sun.com/docs/books/tutorial/networking/urls/readingURL.html

- Lihtsamad brauserid ja brauseritaolised rakendused:
  - lynx
  - wget
  - curl
  - ....

Veebilehtede (failide) serveerimine

- Veebilehed on lihtsalt failid, mille ette server paneb http-päise
- Veebileht (fail) võetakse reeglina kas:
  - olemasoleva failina arvuti kettal
  - tekstina, mille teeb iga kord uuesti mõni programm (cgi)
    - Programm võib olla külge-ehitatud veebiserverile (php, mod-perl jne)
    - ... või olla eraldi programm, mille server käivitab (klassikalised cgi-programmid)

Andmete saatmine serverile: cgi

- URL-i lõpus (peale ?) oleva teksti saab serveri kutsutav programm (nn cgi-programm) kätte
- Brauser saadab vormi info cgi-kodeeritult nii: a=1&b=35 jne, kus a ja b jne on välja nimed, võrduse järel aga kasutaja sisestatud väärtused
- Tüüpiline url cgi parameetritega

http://dijkstra.cs.ttu.ee/~tammet/cgi-bin/loeng/tst6.py?a=1&b=2

http://dijkstra.cs.ttu.ee/~tammet/cgi-bin/loeng/tst8.py?a=12&b=44

- Vt lisaks ja detaile: http://lambda.ee/wiki/Cgi_examples

Brauser oskab mitut keelt

- Iga normaalne brauser mõistab:

Põhiasjad:

- Html
- Css
- Javascript

Lisatehnoloogiad:

- Xslt
- Xml
- Canvas
- Video ja audio
- Väikesed andmebaasid
- ...

Kust lugeda brauseri-tehnoloogiatest

- Ametlikud standardid: http://www.w3.org
- Tutorialid: http://www.w3schools.com
- Tutorialid: http://www.codecademy.com/tracks/web

HTML

- HTML:
  - **Teksti paigutamise / lehe kujundamise keel**
  - Näited kohapeal
  - Loe: http://www.w3schools.com/html/default.asp
  - Eelmine ametlik standard: http://www.w3.org/TR/html401/
  - Uus ametlik standard on **HTML5**: komplekt uusi kasulikke tehnoloogiaid

- Mis on XHTML:
  - Praktiliselt seesama, mis HTML (samad tagid jne)
  - Tagid peavad olema väikeste tähtedega
  - Peab olema korrektses XML süntaksis (tagi suletud jne)
  - Paar XML-i lisaknihvi ka (namespaced jne)
  - Ei ole eriti mainstream või eriti laialt kasutatud

CSS

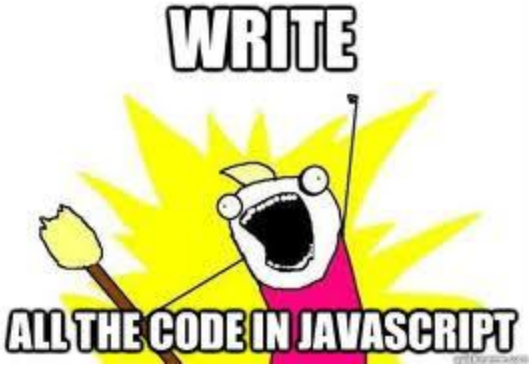
- **Eriti täpset teksti paigutust ja kujundust võimaldav keel HTML täienduseks**

- Näited kohapeal
- Uuri lisaks: http://www.w3schools.com/css/default.asp

Javascript

- **Brauseri programmeerimiskeel: javascripti programmid töötavad otse brauseris: muudavad htmli, css-i, võtavad ühendust serveriga jne jne**

- Näited kohapeal
- Uuri lisaks: http://www.w3schools.com/js/default.asp



Javascript, CSS, asynchronous queries

- **DHTML** tähistas: HTML+CSS+Javascript
- **AJAX** tähistab: HTML+CSS+Javascript+async. Queries
- Vaatame näiteid:
  - http://www.papermountain.org/demos/live/#
  - http://www.sightsmap.com

XML tähendab ...

- **eXtensible Markup Language**

<tootja><nimi>A Le Coq</nimi><linn>Tartu</linn></tootja>

- XML on:
  - Struktureeritud teksti esitamise formaat
  - XML standard ütleb, kuidas teksti struktuuri märgistada.
  - Saab kasutada andmete esitamiseks tekstina
  - Lihtne, aga veidi kohmakas kasutada

- XML ei ole:
  - Programmeerimiskeel.

JSON tähendab ...

- **Javascript Object Notation**

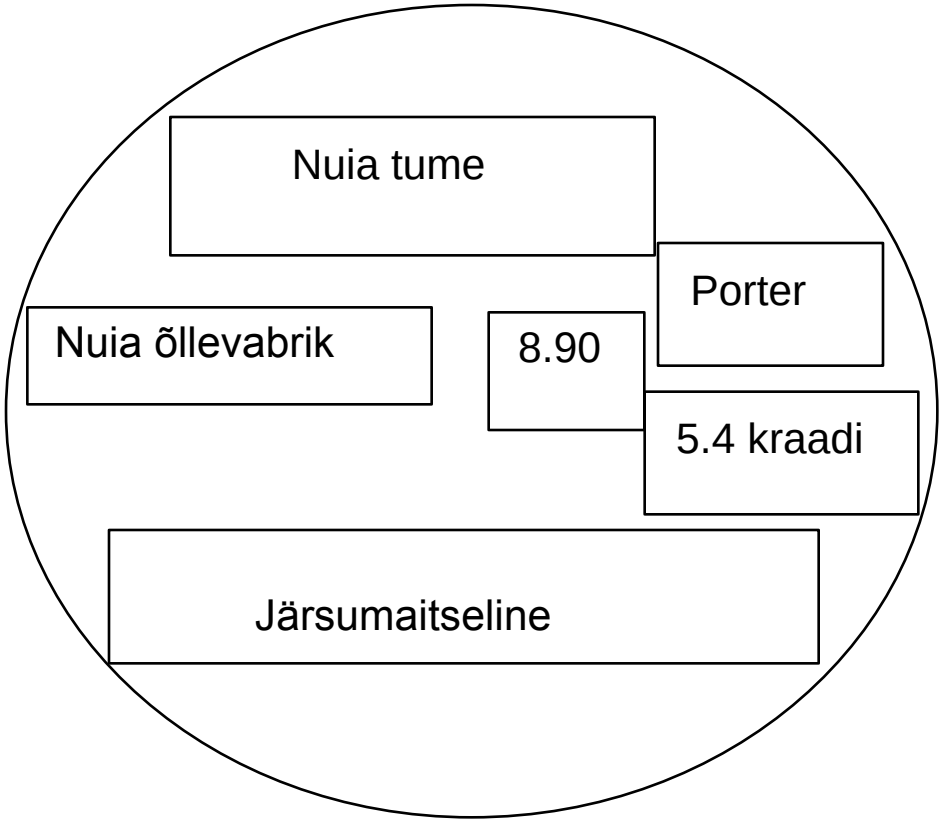
[1,2,“siin on tekst“, [“sisemine“,5], 10]
{“nimi“:“Peeter“, “pikkus“: 180, “hinded“: [5,3,2] }

- JSON on:
  - Andmete esitamise formaat tekstina
  - Javascripti programmeerimiskeele „native“ andmestruktuur
  - Väga lihtne ja mugav kasutada
  - Kaasajal brauserirakendustes eelistatum kui XML.

- JSON ei ole:
  - Programmeerimiskeel.

Tüüpiline probleem andmete esitamisel

- **Meil on mingid andmed.**
- **Neid on vaja faili kirja panna ja hiljem arvutiga töödelda ja teistele edasi saata.**
- Olgu meil info mingi õllesordi kohta:



Kuidas seda faili kirja panna?

- **Variant 1: inimkeelsena.**

  Nuia tumeda tootja on Nuia õllevabrik. Porteri tüüpi õlu. Kangus 5.4 kraadi. Järsumaitseline. Hind 8.90.

  - Oleks hea variant, aga arvutiga praktiliselt lootusetu töödelda.

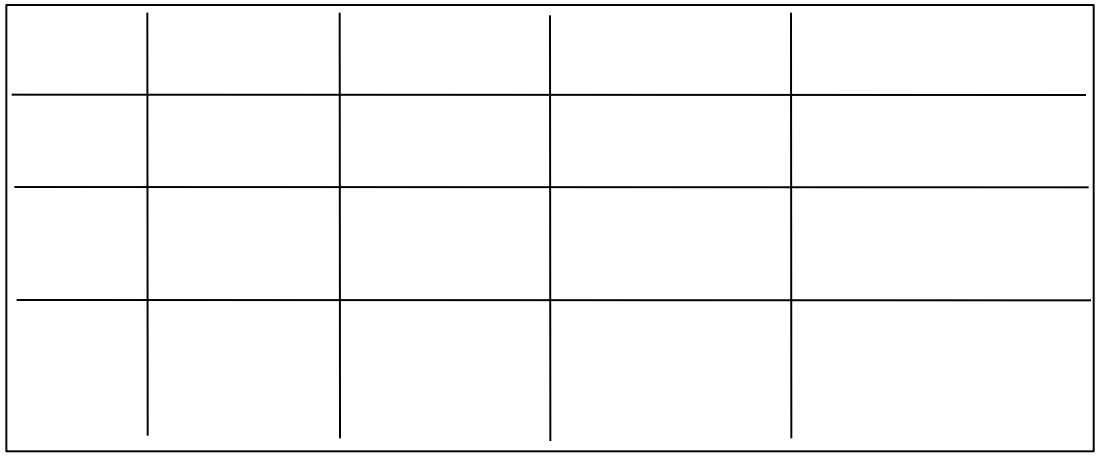
- **Variant 2: CSV (comma separated values)**

  “Nuia tume”, “Nuia õllevabrik”, Porter, 8.90, 5.4, Järsumaitseline

  - Päris hea ja väga levinud variant.
  - Failis ei saa öelda, mis tähestikus (õ, ä jne kodeering).
  - Vaja on eraldi öelda, mida väljad tähendavad (selleks saab kasutada päiserida).
  - Põhiprobleem: järgmisel slaidil

CSV ja muude tabelkujude põhiprobleem

- **CSV ja muud tabeliformaadid esitavad infot tabelina:**



- See ei sobi hästi, kui tabeli ruutudes peaks olema alamtabelid.
- Näiteks, kui:
  - Tahaksime sisestada aadressi: riik, linn/küla, tänav, maja/krt
  - Hinna juures on hinnaühik (EEK, EUR, jne)
  - Jne
- SQL andmebaasides kasutatakse siis abitabeleid ja neid siduvaid kunstlikke ID-numbreid, mis pole seotud algse dataga.

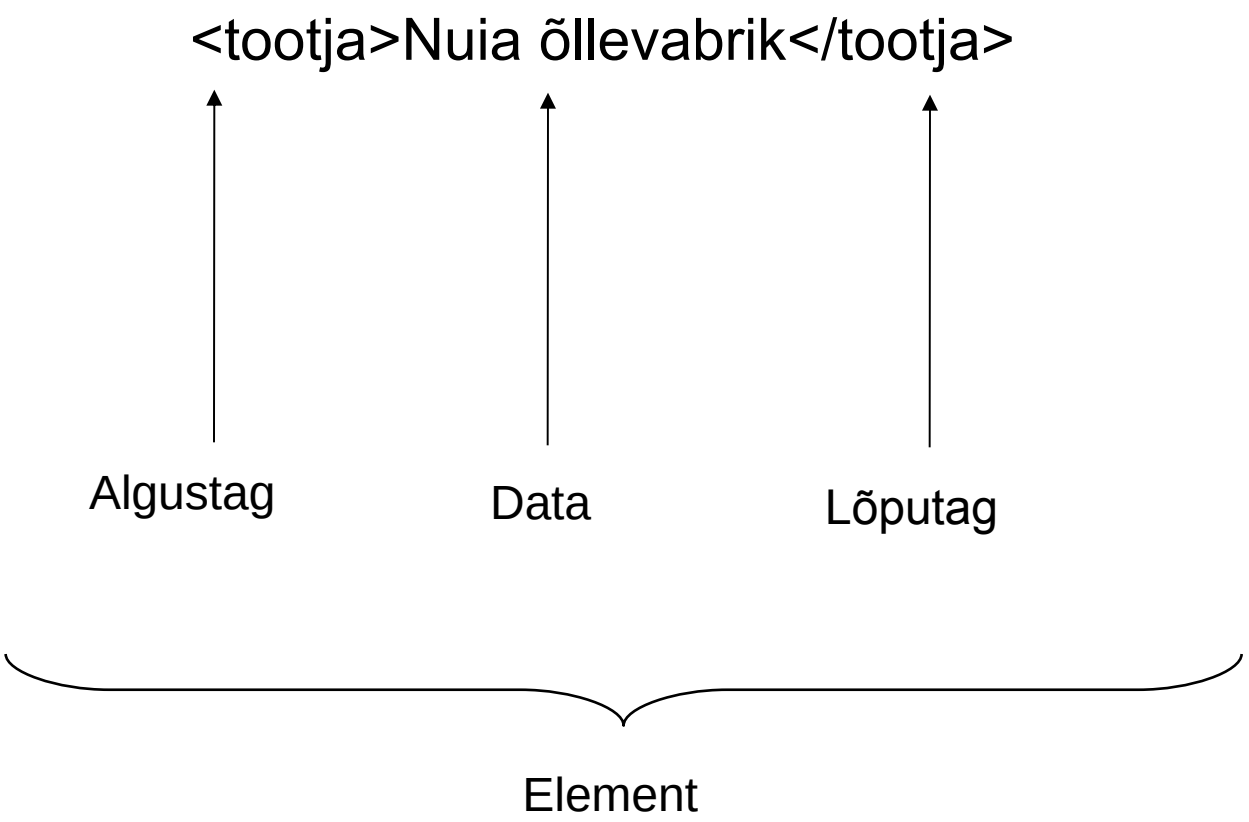
XML viis eelmist infot kirja panna

- **XML-s “pakitakse” infoväljad kirjeldavate nn tag**-de vahele:

```
<olu>
  <nimi>Nuia tume</nimi>
  <tootja>Nuia õllevabrik</tootja>
  <tyyp>Porter</tyyp>
  <hind>8.90</tyyp>
  <kangus>5.4</kangus>
  <maitse>Järsumaitseline</maitse>
<olu/>
```

- Pluss: saab töödelda arvutiga nagu CSV-d
- Pluss: inimesele kohmakas lugeda, kuid siiski üheselt arusaadav.
- Miinus: tekst on pikem, kui CSV puhul.
  - Sellest saab üle: kokkusurumine (zip, gz, jms pakkijad).

XML tagid


<tootja>Nuia õllevabrik</tootja>
Algustag
Data
Lõputag
Element

Tagide põhieelis

- Elemendid võivad sisaldada teisi elemente:

```
<olu>
    <nimi>Nuia tume</nimi>
    <tootja>
        <nimi>Nuia õllevabrik</nimi>
        <aadress>
            <küla>Nuia</küla>
            <tänav>Tallinna mnt</tänav>
            <maja>2</maja>
        </aadress>
    </tootja>
    <tyyp>Porter</tyyp>
    <hind>8.90</tyyp>
    <kangus>5.4</kangus>
   <maitse>Järsumaitseline</maitse>
<olu/>
```

HTML ja XML

- Info pannakse “tag”-ide vahele: infol on sildid

- HTML: “tag”-del visuaalne semantika (st tähendus)

  ****Siin on paks tekst****
  ****Siin on kaldkirjas tekst****

- XML: “tag”-del semantika puudub

  <autor>Tanel Tammet**</autor>**
  <minuaadress>Kuiv 9**</minuaadress>**

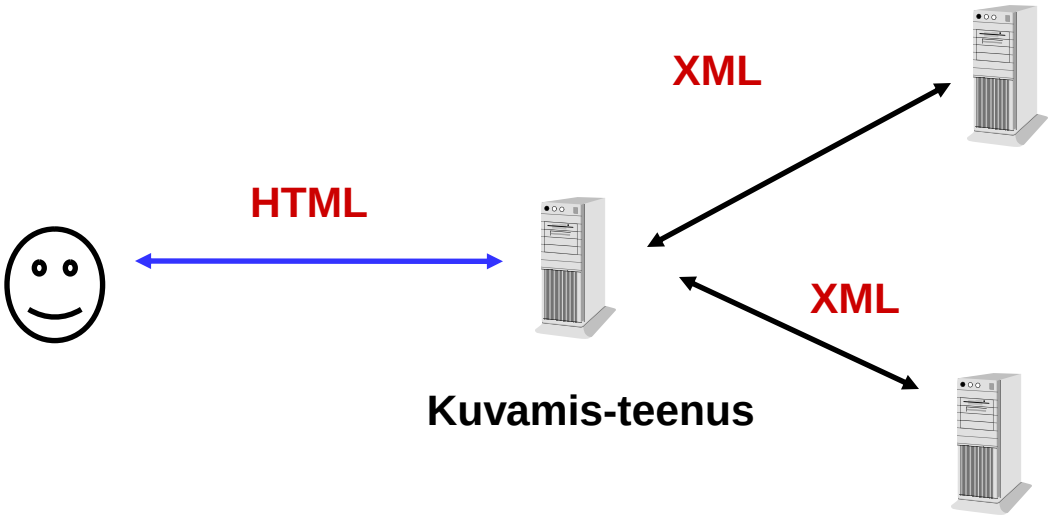
XML vs JSON

- Eelmisel kümnendil muutus XML väga populaarseks andme-esituskeeleks.
- Seejärel hakati kasutama hulka keerulisi lisatehnoloogiad: XSLT, SOAP, XML Schema, UDDI jne, mis muutsid asja keeruliseks.
- Seejärel muutus populaarseks JSON kui hulga lihtsam andme-esituskeel.
- Praegu kasutatakse mõlemat, aga JSON on populaarsem. Laiatarbe-süsteemid a la Google pakuvad enamasti JSON-is andme-esitust

Võrguteenused

- Probleem: raske on teha programme, mis loeks teisest serverist HTML-lehekülgi ja leiaks nende sisust kergesti vajaliku info

- Lahendus: teeme võrgulehekülgi lihtsalt andmetekstina XML-is või JSON-is nii et programm teises arvutis suudaks neid lugeda



Semantic web: w3c suur projekt

- Eesmärk: pakkuda andmete esitamise keeli ja luua standardeid, mille abil kirjeldada asju (internetis)
- Soovitav tulemus: saab teha tarkvara, mille abil datat saab kergesti ühest programmist teise saata, nii et info ja mõte kaduma ei lähe.
- Esmajoones vaja:
  - objektide kirjeldamise keel
  - mõistete omavaheliste seoste ja reeglite keel
- Projekti juht: Tim Berners-Lee (http, www ja esimese brauseri autor, w3c juht).

Kihiline süsteem

- Ülevalt alla:

Tõestuste keel

Täisloogika

Ontoloogiad

Ontoloogiakeel (Owl ja tema alamhulgad)

Asjade kirjelduskeel (RDF ja RDFS)

Nimede unikaliseerimine (namespaces)

Süntaks (XML, JSON, turtle, ...)

Transport (http, https, SOAP, tcp/ip ...)

Mis on keeruline osa semantilise veebi projektis?

- Üks programm peab teise programmi antud XML-ist SISULISELT aru saama
- Õllepartiide otsimise server:

Hulgimüüja A annab infot:

```
<beer>
  <name>Guinness</name>
  <price>100</price>
</beer>
```

Hulgimüüja b annab infot:

```
<olu>
   <mark>Guinness</mark>
   <hind>100</hind>
</olu>
```

Hulgimüüja C annab infot:

```
<porter>
  <name>Guinness</name>
  <price>100</price>
</porter>
```

Erinevad keeled

- Peame teadma, et:

BEER	=	ÕLU
PORTER	on ka	BEER

- Ei ole lootust teha “ette valmis” kõiki keeli
- On olemas alamkeeled (üldine toodete keel, õllede keel Eestis, Tshehhis, Hiinas, õllede keel Tartu Õllevabrikus ja Sakus, ...)
- Keeli on vaja omavahel tõlkida
- Tõlkimiseks on vaja reegleid ja elementaarset arusaamist

Nimede unikaliseerimine: XML namespaces

- Idee:
  - iga tagi ette saab kirjutada, mis keeles ta on <keel:tag>
  - keelte nimedeks kasutatakse URI-sid (URL-e)

- Näide:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:ex="http://www.example.org/terms/">
  <rdf:Description rdf:about="http://www.example.org/index.html">
   <ex:creation-date>August 16, 1999</ex:creation-date>
   <ex:language>English</ex:language>
   <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
  </rdf:Description>
</rdf:RDF>
```

Asjade kirjeldamine: RDF

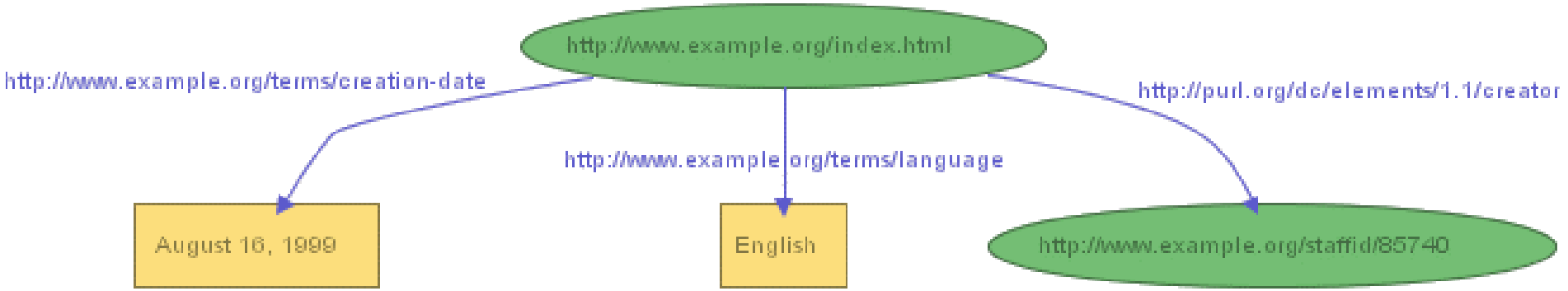
- Idee:
  - Kõik kirjeldused on kolmikud: [objekt, omadus, väärtus]
  - Predikaadina: omadus(objekt,väärtus)
  - Kõik need kolm võivad olla URI-d

kirjelduskeel RDF

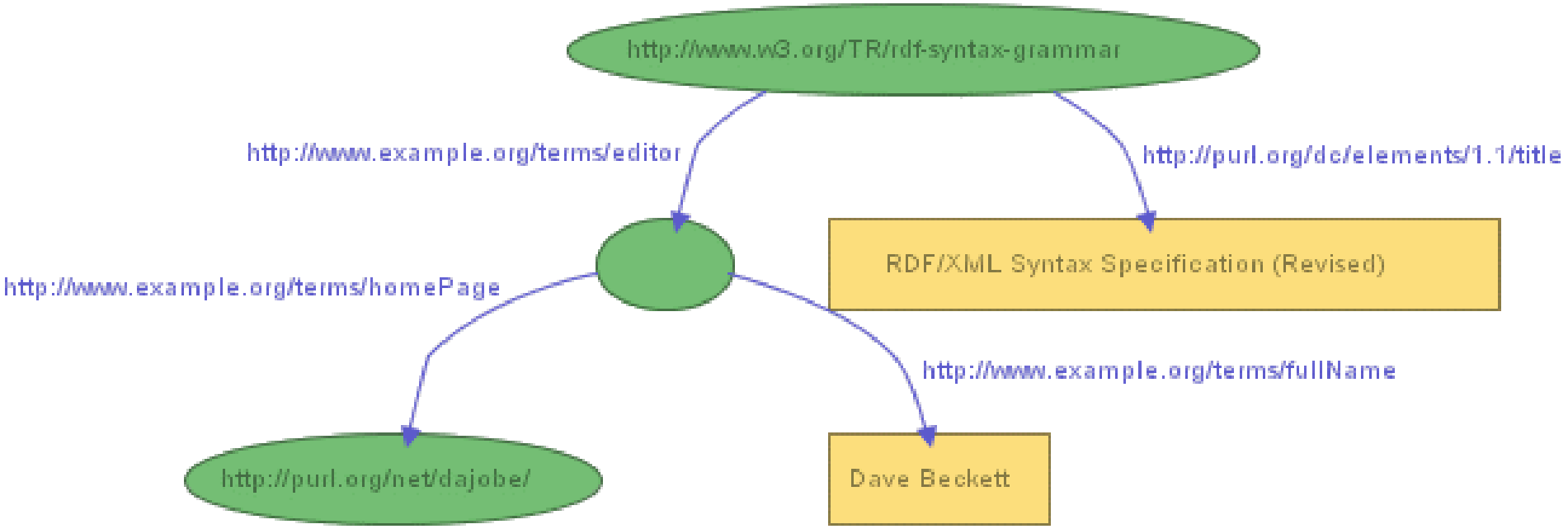
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:ex="http://www.example.org/terms/">
  <rdf:Description rdf:about="http://www.example.org/index.html">
   <ex:creation-date>August 16, 1999</ex:creation-date>
   <ex:language>English</ex:language>
   <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
  </rdf:Description>
</rdf:RDF>
```

Kirjelduskeel RDF

- http://www.example.org/index.html has a creation-date whose value is August 16, 1999
- http://www.example.org/index.html has a language whose value is English
- http://www.example.org/index.html has a creator indicated by http://www.example.org/staffid/85740



kirjelduskeel RDF: sügavam puu



Võrguteenused, RDFa ja microformats

- Üks praktiline viis edasiminekuks semantilise koosvõime osas

  Soovitus esitada andmeid veebilehel selliselt, et:

  - andmed oleks harilikul (suvalisel) inimloetaval html-kujul tekstidena. xml formaat ei ole tingimata vajalik.
  - harilikul html-lehel olevate andmejuppide ümber pannakse RDFa formaadis metainfo, mis ütleb automaattöötlejale, mis andmetega on tegemist
  - lisatav metainfo ei muuda lehe algset väljanägemist ja üldjuhul ei dubleeri lehel juba olevat infot

Võrguteenused ja RDFa

- Näide 1

Algse html-teksti

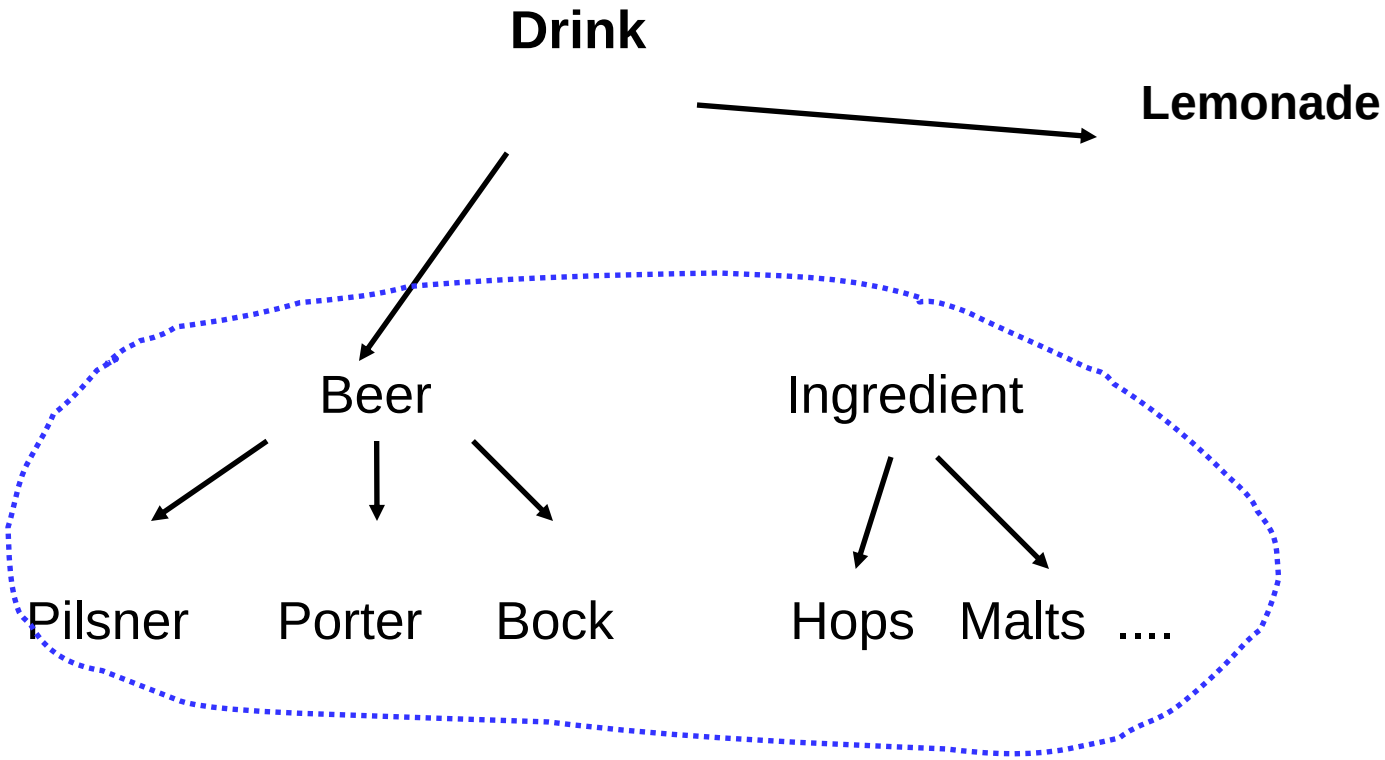
```
<tr>
   <td>Jaanus Kask</td>
   <td>6024554</td>
   <td><a href="http://kask.googlepages.com">kliki siia</a></td>
</tr>
```

asemel on html-tekstiks RDFa järgi annoteeritult

```
<tr about="#jaanus_kask">
   <td property="er:nimi">Jaanus Kask</td>
   <td property="er:telefon">6024554</td>
   <td><a rel="er:koduleht"
          href="http://kask.googlepages.com">kliki siia</a></td>
</tr>
```

Ontoloogiad

- Mis on ontoloogiad: “maailma mudelid” ehk mõistete hierarhiad



- Eri valdkondades on oma ontoloogiad

- Ilmselt ei õnnestu panna kõiki programme kasutama samu “standardontoloogiaid”